Speaker-Independent Speech Recognition using Visual Features

Pooventhiran G.[1], Sandeep A.[2], Manthiravalli K.[3], Harish D.[4], Karthika Renuka D.[5]
Undergraduate, Dept. of Information Technology
PSG College of Technology
Coimbatore, Tamil Nadu, India - 641004[1,2,3,4]
Associate Professor, Dept. of Information Technology
PSG College of Technology
Coimbatore, Tamil Nadu, India - 641004[5]

***Abstract*—Visual Speech Recognition aims at transcribing lip movements into readable text. There have been many strides in automatic speech recognition systems that can recognize words with audio and visual speech features, even under noisy conditions. This paper focuses only on the visual features, while a robust system uses visual features to support acoustic features. We propose the concatenation of visemes (lip movements) for text classification rather than a classic individual viseme mapping. The result shows that this approach achieves a significant improvement over the state-of-the-art models. The system has two modules; the first one extracts lip features from the input video, while the next is a neural network system trained to process the viseme sequence and classify it as text.**

Keywords*—Visual speech recognition; audio speech recognition; visemes; lip reading system; Convolutional Neural Network (CNN)

I. Introduction

Visual Speech Recognition (VSR) is the process of extracting textual or speech data from facial features through image processing techniques. It plays a vital role in human-computer interaction; mostly in noisy environments, it complements Automatic Speech Recognition systems to improve performance [1][2]. Like speech recognition systems, lip reading (LR) systems also face problems due to variances in skin tone, speaking speed, pronunciation, and facial features. A standalone lip reading system may not be very efficient. Several factors, such as skin tone, accents, duration of utterances, limit this efficiency. The LR systems can be synchronized with an Audio Speech Recognition system to improve the confidence of classification by using both model's advantages [3]. Many systems limit the datasets to contain only a few words and phrases rather than all possible sentences to simplify this problem. Speech recognition systems are of two types: Speaker-dependent and Speaker-independent systems. Speaker-dependent systems train on data from a single speaker and are suitable for speech and speaker verification applications [4]. Speaker-independent systems train on data from several speakers to generalize and are suitable for text transcription and voice-activated applications. Our project is a speaker-independent system trained on data from lip movements (lip features or visemes) extracted from the input video file. The input will have many parameters like height, width, and frame rate. Our system emphasizes the same frame rate. It extracts lip features from each frame and stores them. A problem found is that there will not be any perceivable difference between the two frames. Also, a training dataset cannot provide apt text matches when trained with a different number of frames. Thus, we go by concatenating a fixed number of frames and classify a sequence of visemes directly to text rather than to phonemes [5]. The system comprises two segments: one being the feature extraction system that extracts lip features and makes it into a visual feature cube, while the other being a Convolutional Neural Network trained on a rich dataset, which matches visemes to the corresponding text.

The paper is organized as follows: Section 2 explores the related literature; Section 3 describes the dataset used in the experiment; the proposed technique is explained in Section 4, followed by an analysis of results in Section 5 while Section 6 concludes the paper.

II. Existing Models

In VSR systems, only the lip movements provide a significant contribution to knowledge retrieval. Many approaches are used in the literature to extract different features for LR systems.

A. Lip Feature Extraction in YIQ domain

This method proposed by [6] converts the video sequence in the Red Green Blue (RGB) domain to the Luminance Inphase Quadrature (YIQ) domain. The 'Y' component represents the luminance, while 'I' and 'Q' represent the chrominance information. Using the YIQ format helps localize lip features as human lips are usually brighter in the 'Q' space while the overall face is brighter in the 'I' space. A solid model can exploit this contrast for lip localization and lip tracking by segmenting the image in 'I' space.

B. Segmentation Method

In this method, [7] uses two approaches: edge detection and region segmentation. These methods detect the contour of the outer lip, and their results are combined using AND or OR fusion. They first found the mouth Region of Interest (ROI), which is then given to edge detection and region segmentation methods. The combination of results from these two methods provides the final outer lip contour.

C. Zernike Features

The model proposed by [8] aims to improve audio-visual recognition accuracy. The proposed solution includes extracting visual features using Zernike moments and audio features using Mel frequency cepstral coefficients on the visual vocabulary of independent standard words dataset on a series on the visual utterance. 'Viola-Jones' detector based on 'AdaBoost' method, used for face recognition and mouth portion, is calculated from the ROI bounding box's median coordinates. Zernike movements for ROI are computed for each frame resulting in 9x1 columns. One visual utterance is captured for two seconds forming 52 frames; therefore, the Zernike features for one visual utterance result in 468x1 for a single word. Further, Principal Component Analysis (PCA) is used to convert original features to independent linear variables possessing the most information. The performance, which was based on visual-only and audio-only features, resulted in 63.88% and 100% accuracy, respectively, which is relatively higher.

D. Deep Neural Networks

This Speaker-independent lip reading system by [9] uses techniques such as Linear Discriminant Analysis (LDA), Maximum Likelihood Linear Transform (MLLT), and Speaker Adaptive Training (SAT). The visual features are extracted in the following pipeline: the features are mean-normalized on a per-speaker basis and are decorrelated and reduced to a dimension of 40 using LDA and MLLT, and then, SAT is applied to normalize the variation in acoustic features of different speakers.

DNN is experimented as promising for speaker-independent lip reading even with limited training data and without a pre-training stage. The best-known result for a speaker-independent lip reading system is to use a hybrid system that uses MLLT followed by SAT.

III. Dataset

In the first module, viseme extraction is done using the DLIB module functions, which uses a pre-trained dataset, while the second module employs CNN that uses MIRACL-VC1 dataset [10].

A. Shape Predictor

MIRACL-VC1 is a trained dataset for dlib used for matching visemes, called "shape predictor 68 face landmarks". It provides the means to match facial features. The interface is provided through predictor and detector classes from the dlib package. The face detector used is made using the classic Histogram of Oriented Gradients (HOG) feature combined with a linear classifier, an image pyramid, and a sliding window detection scheme.

The landmark points from 48 to 68, shown in Fig. 1, are assumed to approximate the lip portion. So, those landmarks are considered as edge points while cropping.

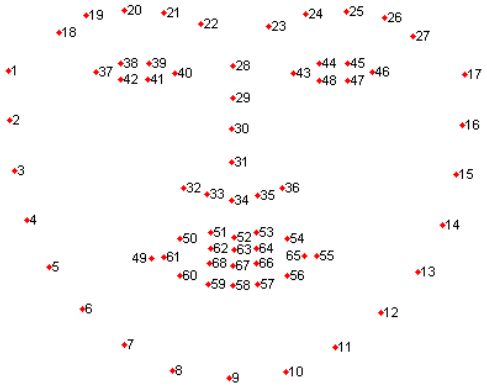

Fig. 1. Template 68-Point Facial Feature Image [11]

TABLE I. Words and Phrases available in MIRACL-VC1 dataset

ID	Word	ID	Phrase
01	Begin	01	Stop navigation
02	Choose	02	Excuse me
03	Connection	03	I am sorry
04	Navigation	04	Thank you
05	Next	05	Good bye
06	Previous	06	I love this game
07	Start	07	Nice to meet you
08	Stop	08	You are welcome
09	Hello	09	How are you
10	Web	10	Have a good time

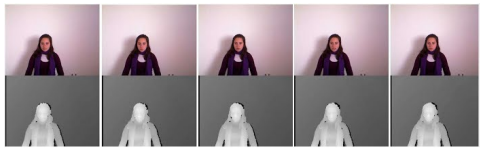

Fig. 2. Sample Color and Depth Image Frames [10]

B. MIRACL-VC1

MIRACL-VC1 is a lip reading database that includes both depth and color images as features. It facilitates multiple research areas such as speech recognition, face detection, and biometrics. Fifteen speakers (five men and ten women) who are positioned in the view of a Microsoft Kinect sensor utter ten times a set of ten words and ten phrases as shown in Table I. Each example in the dataset comprises color and depth images, both of size 640x480, synchronized. The sample color and depth images are shown in Fig. 2. The dataset contains a total number of 3000 examples (15 x 10 x 10 = 1500 images - color and depth images each). Our system utilizes only color images of words.

IV. Design and Implementation

The first step is extracting lip features from the video. The features are further given to a 3D CNN [4] that can classify visemes to the corresponding text. These two functionalities are separated into two modules: the pre-processing module and the CNN module. The input video file is pre-processed to extract lip features from the facial features and fed to CNN to

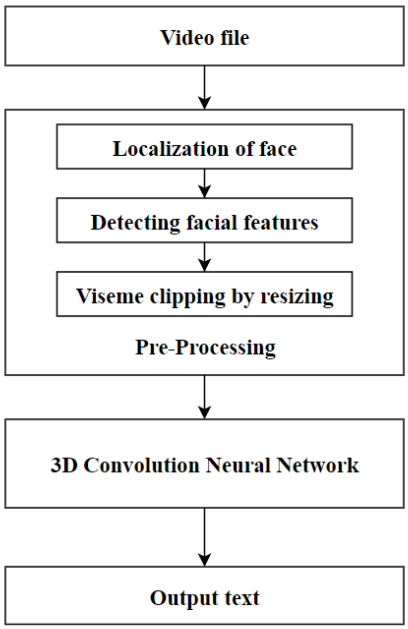

Fig. 3. Flow Chart of the Proposed Design

classify the visemes to text.

The flowchart shown in Fig. 3 is explained as follows.

A. Pre-Processing

The visemes need to be extracted from each frame. For this, the video is broken into individual frames first. Since the frame rate differs from one video to another, they need to be equalized to have the same frame rate (30fps). The processed frames are passed to the face tracking module.

B. Face Tracking

Facial tracking obtains data about still images and video sequences by automatically tracking the facial landmarks. Specific facial landmarks mapped, such as 48 face landmarks, 68 face landmarks, are available. It involves two steps: Localization of face and Detection of key facial structures. Since we do not need all the points in the frame, only the facial region is tracked first and using 68 facial landmarks, the key features are detected. We use the frontal face detector and shape predictor modules of the dlib package to achieve this.

1) Localization of Face: A pre-trained Histogram of Oriented Gradients (HOG) with Linear SVM Object Detector or deep learning-based algorithms can be applied to localize the face. The aim is to obtain the (x, y) coordinates of the face (formed as a bounding box) through these methods.

2) Detection of Key Facial Structures: A variety of facial landmark detectors are available that try to localize and label the following facial regions effectively:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

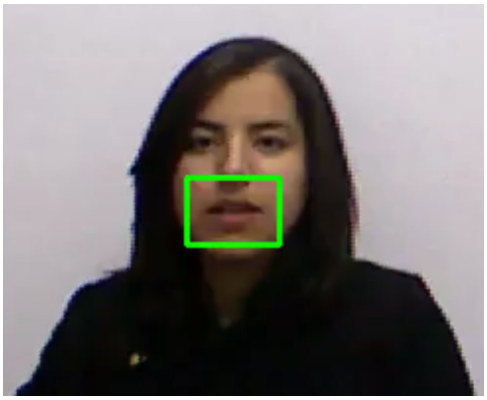

Fig. 4. Lip Region Bordered Input Video

The dlib library uses One Millisecond Face Alignment with an Ensemble of Regression Trees face detector from Kazemi and Sullivan [12]. The method works using the following:

1) The image is projected into a normalized coordinate system from which features are extracted. This process is repeated until convergence.
2) Prior probabilities on the distance between pairs of input pixels to boost the algorithm to work efficiently on a large number of relevant features.

The method builds an ensemble of regression trees on the training data to estimate the facial landmark positions by identifying pixel intensities that correspond to these landmarks themselves. This library, coupled with OpenCV, can provide a detector that can capture the necessary points, in our case, the lip visemes coordinates, as shown in Fig. 4.

C. Resizing

The tracked facial images can be of any angle of view. The speaker would have spoken the phrases either by looking straight into the camera or while looking somewhere. Since this poses a difficulty in detecting facial features, as some features may be lost, we can either restrict the speakers where to look or cropping and resize the image only to contain the lip region. The face images may be in different sizes. So, the detected faces are clipped and resized into the same size (30x48), which helps the CNN process them efficiently. It can be done by finding out the lip region edge points and cropping the image's desired portion. The resized images are shown in Fig. 5.

D. Convolutional Neural Network

Convolutional Neural Networks are most commonly used for image processing tasks [13]. The CNN architecture used is shown in Fig. 6. The model inputs a sequence of visemes (gray-scale) of dimension 15x30x48x1. The input is processed

Fig. 5. Cropped and Resized Visemes

```
---3D CNN for Visual Speech Recognition---
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1 (Conv3D)               (None, 13, 28, 46, 8)     224
_________________________________________________________________
batch_normalization (BatchNo (None, 13, 28, 46, 8)     32
_________________________________________________________________
activation (Activation)      (None, 13, 28, 46, 8)     0
_________________________________________________________________
conv2 (Conv3D)               (None, 11, 26, 44, 16)    3472
_________________________________________________________________
batch_normalization_1 (Batch (None, 11, 26, 44, 16)    64
_________________________________________________________________
activation_1 (Activation)    (None, 11, 26, 44, 16)    0
_________________________________________________________________
conv3 (Conv3D)               (None, 9, 24, 42, 32)     13856
_________________________________________________________________
batch_normalization_2 (Batch (None, 9, 24, 42, 32)     128
_________________________________________________________________
activation_2 (Activation)    (None, 9, 24, 42, 32)     0
_________________________________________________________________
pool1 (MaxPooling3D)         (None, 9, 11, 20, 32)     0
_________________________________________________________________
flatten (Flatten)            (None, 63360)             0
_________________________________________________________________
fc1 (Dense)                  (None, 32)                2027552
_________________________________________________________________
batch_normalization_3 (Batch (None, 32)                128
_________________________________________________________________
activation_3 (Activation)    (None, 32)                0
_________________________________________________________________
dropout (Dropout)            (None, 32)                0
_________________________________________________________________
fc2 (Dense)                  (None, 10)                330
_________________________________________________________________
softmax (Activation)         (None, 10)                0
=================================================================
Total params: 2,045,786
Trainable params: 2,045,610
Non-trainable params: 176
_________________________________________________________________
None
```

Fig. 6. Architecture of CNN used

by three Conv3D layers of 8, 16, and 32 neurons each. This stacking of three layers learns low-level features like edges and lines of the viseme and gradually high-level features such as lip movements and their sequence patterns. A batch normalization and activation layer follow each such layer. The features learned from these layers are sampled down by a max-pooling layer and vectorized using the Flatten layer. The features learned are then passed to a fully-connected layer of 32 neurons with L2 regularization, followed by batch normalization and activation layers, and given to the softmax classifier of 10 neurons matching the number of output classes. Table II shown below, presents the hyperparameters used in the CNN architecture.

V. Result Analysis

This paper uses accuracy to evaluate the experiment, along with precision, recall, and F-measure metrics. Eq. 1, 2, 3, and 4, respectively show the formulae for computing these metrics.

$$Accuracy\& = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision\& = \frac{TP}{TP + FP} \tag{2}$$

TABLE II. Hyperparameters for the CNN model

Parameter	Value
Kernel size	3x3x3
Stride	1x1x1
Pool size	1x3x3
Activation	ReLU
Optimizer	SGD
Learning rate	1e-2
Regularization factor	1e-2
Dropout factor	20%
Batch size	32

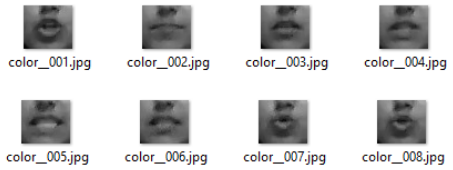

Fig. 7. Visemes of the Second Utterance of 'Choose'

TABLE III. Class-wise metrics

Metrics & Words	Precision (%)	Recall (%)	F-measure (%)
Begin	85.71	88.89	87.27
Choose	90.48	73.08	80.85
Connection	90.48	76.0	82.61
Navigation	78.26	75.0	76.6
Next	84.62	91.67	88.0
Previous	71.43	83.33	76.92
Start	73.68	70.0	71.79
Stop	87.5	60.87	71.79
Hello	45.95	89.47	60.71
Web	80.0	64.0	71.11
Weighted avg.	**80.24**	**76.89**	**77.40**

$$Recall\& = \frac{TP}{TP + FN} \tag{3}$$

$$F - measure\& = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

We concatenate the frames for each word to form a training example. Since the number of frames is different for each word due to utterance duration variation, we fixed the number of frames to 15 and padded the sequence with fewer than 15 frames with a viseme for a closed mouth. This padding method represents humans' closed-mouth position while we are not speaking, facilitating more human-like processing. The frames fewer than 15 for the word "Choose" are shown in Fig. 7.

Fig. 8 presents the confusion matrix obtained for the proposed model. This matrix shows that the model can robustly classify the viseme sequence to the target text. Table III lists the metrics class-wise. The F-measure for the model also shows that the classifier is more generalized and not biased towards any class.

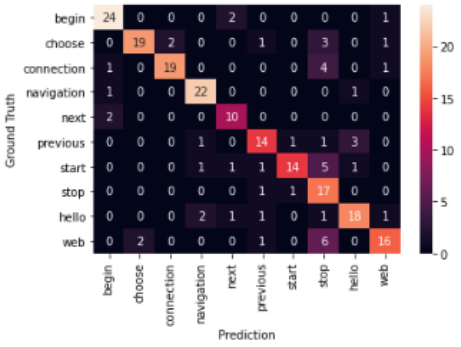

Fig. 8. Confusion Matrix

TABLE IV. COMPARISON OF THE PROPOSED METHODS WITH STATE-OF-THE-ART METHODS

Model	Accuracy
Borde et al., [8]	63.88%
Garg et al., [14]	56%
Proposed model	**76.89%**

Table IV compares the accuracy of various state-of-the-art models with the proposed model. Our model achieves about 76.89% accuracy, which is a significant improvement over the state-of-the-art models.

VI. CONCLUSION

This paper presented a combined approach of visemes concatenation and 3D Convolutional Neural Networks for Speaker-independent Visual Speech Recognition. We used dlib's face detection module to localize the face features in each frame, and with the help of 68-facial landmarks, we extracted the lip portion. The extracted visemes are cropped and resized to avoid them from being at different angles, improving the classifier's performance. We concatenated these frames of each word to generate an input feature. To fix the variation in the number of frames due to each word's utterance duration, we fixed the number of frames at 15. The 3D CNN learns from the sequence of visemes, the pattern for each word. The low-level and high-level features are appropriately learned from the hidden CNN layers. Our experiment shows that this approach outperforms the state-of-the-art models by improving the classification accuracy.

REFERENCES

[1] A. Thanda and S. M. Venkatesan, "Audio visual speech recognition using deep recurrent neural networks," in *IAPR workshop on multimodal pattern recognition of social signals in human-computer interaction*. Springer, 2016, pp. 98–109.

[2] J. Wang, J. Zhang, K. Honda, J. Wei, and J. Dang, "Audio-visual speech recognition integrating 3d lip information obtained from the kinect," *Multimedia Systems*, vol. 22, no. 3, pp. 315–323, 2016.

[3] E. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke, "An improved automatic lipreading system to enhance speech recognition," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1988, pp. 19–25.

[4] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson, "3d convolutional neural networks for cross audio-visual matching recognition," *IEEE Access*, vol. 5, pp. 22 081–22 091, 2017.

[5] N. Alothmany, R. Boston, C. Li, S. Shaiman, and J. Durrant, "Classification of visemes using visual cues," in *Proceedings ELMAR-2010*. IEEE, 2010, pp. 345–349.

[6] T. N. Sengupta, "S.: Lip localization and viseme recognition from video sequences," in *Fourteenth National Conference on Communications*, 2008.

[7] U. Saeed and J.-L. Dugelay, "Combining edge detection and region segmentation for lip contour extraction," in *International Conference on Articulated Motion and Deformable Objects*. Springer, 2010, pp. 11–20.

[8] P. Borde, A. Varpe, R. Manza, and P. Yannawar, "Recognition of isolated words using zernike and mfcc features for audio visual speech recognition," *International journal of speech technology*, vol. 18, no. 2, pp. 167–175, 2015.

[9] I. Almajai, S. Cox, R. Harvey, and Y. Lan, "Improved speaker independent lip reading using speaker adaptive training and deep neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2722–2726.

[10] A. Rekik, A. Ben-Hamadou, and W. Mahdi, "A new visual speech recognition approach for rgb-d cameras," in *International conference image analysis and recognition*. Springer, 2014, pp. 21–28.

[11] P. Huber, *Real-time 3D morphable shape model fitting to monocular in-the-wild videos*. University of Surrey (United Kingdom), 2017.

[12] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[13] S. L. Rose, L. A. Kumar, and D. K. Renuka, "Deep learning using python," 2019.

[14] A. Garg, J. Noyola, and S. Bagadia, "Lip reading using cnn and lstm," *Technical report, Stanford University, CS231 n project report*, 2016.